

# **Symulacja Cyfrowa**

## **Projekt**

**Oskar Czyżewski**

**numer albumu: 126142**

**numer zadania: D3 S2 M3**

**metoda symulacji: ABC**

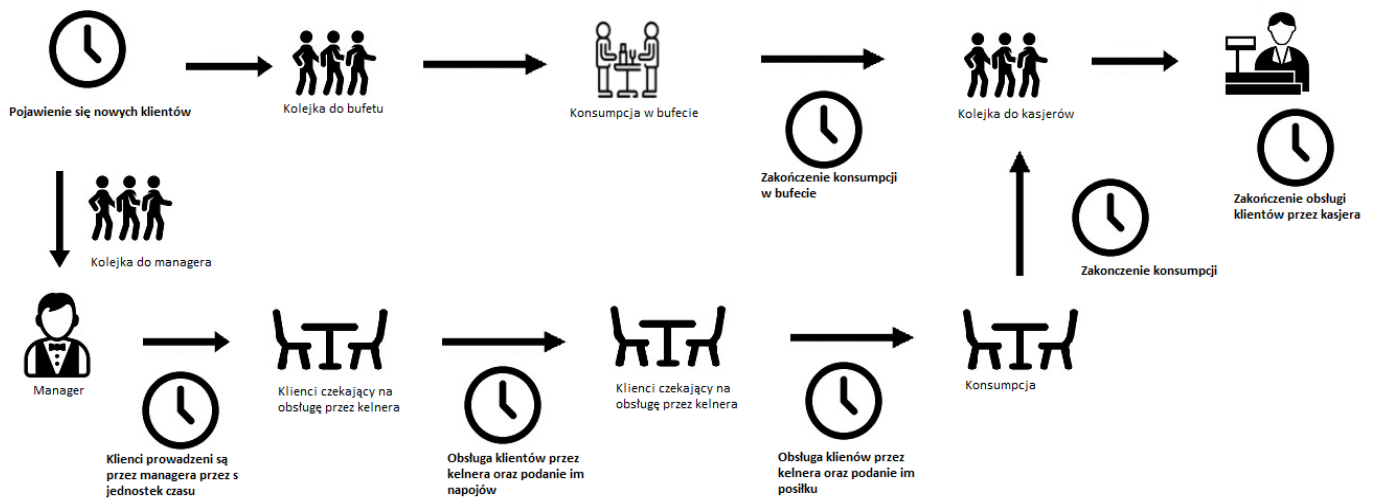
## 1. Treść zadania

W chińskiej restauracji pracuje  $k$  kelnerów obsługujących  $n_2$  stolików dwuosobowych,  $n_3$  stolików trzyosobowych oraz  $n_4$  stolików czteroosobowych. Klienci pojawiają się w restauracji jako grupy 1-, 2, 3- lub 4-osobowe z prawdopodobieństwami odpowiednio  $p_1$ ,  $p_2$ ,  $p_3$  oraz  $p_4$ . Odstęp czasu rozdzielający pojawienie się kolejnych grup klientów jest zmienną losową o rozkładzie normalnym ze średnią  $\mu_a$  i wariancją  $\sigma_a^2$ . Jeśli jest dostępny stół odpowiadający wielkości grupy (lub większy), klienci są do niego prowadzeni przez kierownika sali (czynność ta zajmuje  $s$  jednostek czasu). W przeciwnym przypadku grupa oczekuje na stół w kolejce. Średnio połowa klientów korzysta z samoobsługowego bufetu, przy którym może znajdować się jednocześnie  $b$  osób. Czas spędzany przy bufecie przez grupę klientów jest zmienną losową o rozkładzie normalnym ze średnią  $\mu_b$  i wariancją  $\sigma_b^2$ . Pozostali klienci są obsługiwani przez tego z kelnerów, który jako pierwszy będzie wolny. W pierwszej kolejności klienci otrzymują napoje, a następnie serwowane jest danie główne. Czas obsługi w obu przypadkach jest zmienną losową o rozkładzie wykładniczym ze średnimi odpowiednio  $\lambda_n$  oraz  $\lambda_j$  (te dwie wielkości uwzględniają zarówno czas oczekiwania na zrealizowanie zamówienia jak i sam czas podania napojów i posiłku głównego). Po zakończeniu konsumpcji, której długość jest zmienną losową o rozkładzie wykładniczym ze średnią  $\lambda_f$ , klient płaci jednemu z  $c$  zatrudnionych kasjerów. Czas obsługi przez kasjera jest zmienną losową o rozkładzie wykładniczym ze średnią  $\lambda_p$ . W restauracji zamontowano nieprawidłowo system przeciwpożarowy. Co jakiś czas, bez przyczyny, rozlega się dźwięk alarmu. Część gości, świadoma nieprawidłowości, pozostaje na miejscu, natomiast reszta opuszcza restaurację. Odstęp czasu rozdzielający kolejne alarmy jest zmienną losową o rozkładzie normalnym ze średnią  $\mu_e = 4200$  i wariancją  $\sigma_e^2 = 50^2$ . Prawdopodobieństwo, że dana grupa nie opuści restauracji wynosi 70%. Zakładając, że kierownik sali zawsze wybiera stół najlepiej pasujący do danej grupy oraz stosuje jedną z podanych niżej zasad obsługi kolejki, oszacuj za pomocą odpowiedniego eksperymentu symulacyjnego:

- średni czas oczekiwania na stół,
- średnią długość kolejki oczekujących na stół,
- średni czas oczekiwania na obsługę przez kelnera od momentu zajęcia miejsca przy stole,
- średnią długość kolejki przy kasach.

## 2. Opis modelu symulacyjnego

Schemat modelu symulacyjnego



## Opis klas wchodzących w skład systemu i ich atrybutów

Nazwa Klasy	Opis	Atrybuty
Restaurant	Klasa reprezentująca restaurację. Zawiera wszystkie kolejki i atrybuty potrzebne do symulacji.	<p><b>-waiters_</b> - zawiera liczbę kelnerów w restauracji</p> <p><b>- table_two_</b> - zawiera liczbę stolików dwuosobowych w restauracji</p> <p><b>-table_three_</b> - zawiera liczbę stolików trzyosobowych w restauracji</p> <p><b>- table_four_</b> - zawiera liczbę stolików czterosobowych w restauracji</p> <p><b>- buffet_seats_</b> - zawiera liczbę miejsc w bufecie</p> <p><b>- manager_</b> - flaga bool sprawdzająca czy kierownik restauracji prowadzi klientów do stolika</p> <p><b>cashiers_</b> - zawiera liczbę kasjerów w restauracji</p> <p><b>queue_for_table_</b> - reprezentuje kolejkę klientów do kierownika</p> <p><b>queue_for_buffet_</b> - reprezentuje kolejkę klientów do bufetu</p> <p><b>queue_for_cashiers_</b> - reprezentuje kolejkę klientów do kasjera</p> <p><b>queue_for_drink_</b> - reprezentuje kolejkę klientów do kelnera po napoje</p> <p><b>queue_for_meal_</b> - reprezentuje kolejkę klientów do kelnera po jedzenie</p> <p><b>buffet_members_</b> - reprezentuje grupę klientów znajdujących się w bufecie</p> <p><b>clients_group_drinks_</b> - reprezentuje grupę klientów obsługiwanych przez kelnera ( podanie napojów )</p> <p><b>clients_group_meals_</b> - reprezentuje grupę klientów obsługiwanych przez kelnera ( podawanie jedzenia )</p> <p><b>members_consumption_</b> reprezentuje grupę klientów konsumujących jedzenie</p>
Event	Klasa po której dziedziczą wszystkie zdarzenia czasowe wykorzystywane w symulacji	<p><b>event_time_</b> - reprezentuje symulacyjny czas występowania zdarzenia</p> <p><b>flag_</b> - reprezentuje zmienną bool, która jest przypisywana do każdego zdarzenia czasowego. Za jej pomocą weryfikuje się czy zdarzenie ma być usunięte z listy zdarzeń</p> <p><b>system_time_</b> - reprezentuje czas systemowy</p>

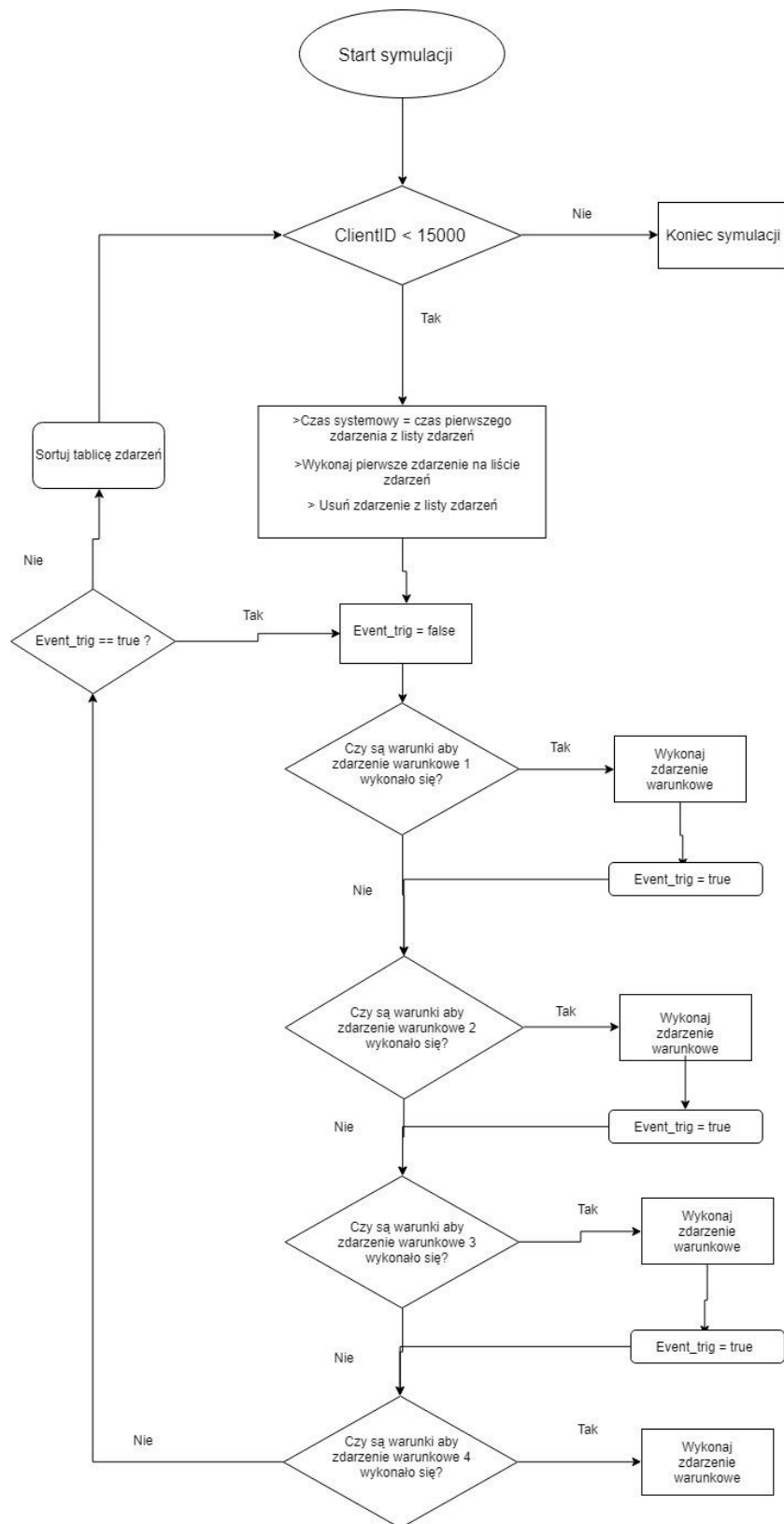
		<b>id_</b> - służy do ustawiania ID kolejnym wchodzącym klientom do restauracji <b>manager_time_</b> - reprezentuje systemowy czas prowadzenia klienta przez managera
Events	Klasa zawierająca metody wykonujące zdarzenia warunkowe	
Generators	Klasa służy do generowania liczb pseudolosowych. Zawiera w sobie implementację generatorów o rozkładzie normalnym, wykładniczym i równomiernym	<b>avg_a_</b> - parametr $\mu_a$ <b>sigma_a_</b> - parametr $\sigma_a$ <b>avg_b_</b> - parametr $\mu_b$ <b>sigma_b_</b> - parametr $\sigma_b$ <b>lambda_n_</b> - parametr $\lambda_n$ <b>lambda_j_</b> - parametr $\lambda_j$ <b>lambda_f_</b> - parametr $\lambda_f$ <b>lambda_p_</b> - parametr $\lambda_p$ <b>avg_e_</b> - parametr $\mu_e$ <b>sigma_e_</b> - parametr $\sigma_e$ <b>x_</b> - ziarno
Statistic	Klasa generująca statystyki	Atrybuty do generowania:  Średniego czasu oczekiwania na stolik <b>avg_wait_for_table_;</b> <b>count_wait_for_table_;</b> <b>n_wait_for_table_;</b>  Średniej długości kolejki do stolików <b>avg_queue_table_length_;</b> <b>count_queue_table_length_;</b> <b>n_queue_table_length_;</b>  Średniego czasu oczekiwania na kelnera <b>avg_waiter_;</b> <b>count_waiter_;</b> <b>n_waiter_;</b>  Średniej długości kolejki do kasjera <b>avg_queue_cashier_length_;</b> <b>count_queue_cashier_length_;</b> <b>n_queue_cashier_length_;</b>  <b>phase_</b> - definiuje czy faza początkowa już się zakończyła

Client	Klasa reprezentująca grupę klientów przechodzącego przez kolejne elementy restauracji	<b>client_id_</b> - unikalny atrybut każdego klienta przypisywany przy wejściu do restauracji <b>number_people_</b> - reprezentuje ilość osób w grupie <b>time_to_table_</b> - reprezentuje czas oczekiwania klienta na stolik <b>time_to_waiter_</b> - reprezentuje czas oczekiwania klienta na kelnera <b>aware_</b> - zmienna boolowska reprezentująca czy klient ma świadomość wadliwości alarmu <b>table_</b> - reprezentuje wielkość stolika przy którym siedzi grupa klientów <b>consumption</b> – używana do sortowania klientów przy zdarzeniach czasowych
Alarm	Klasa reprezentująca zdarzenie czasowe: rozleganie się dźwięku alarmu	<b>event_time_</b> - reprezentuje symulacyjny czas występowania zdarzenia <b>flag_</b> - reprezentuje zmienną bool, która jest przypisywana do każdego zdarzenia czasowego. Za jej pomocą weryfikuje się czy zdarzenie ma być usunięte z listy zdarzeń
CashierEvent	Klasa reprezentująca zdarzenie czasowe: koniec obsługi klienta przez kasjera	<b>event_time_</b> - reprezentuje symulacyjny czas występowania zdarzenia <b>flag_</b> - reprezentuje zmienną bool, która jest przypisywana do każdego zdarzenia czasowego. Za jej pomocą weryfikuje się czy zdarzenie ma być usunięte z listy zdarzeń
EndBuffet	Klasa reprezentująca zdarzenie czasowe: koniec konsumpcji klienta w bufecie	<b>event_time_</b> - reprezentuje symulacyjny czas występowania zdarzenia <b>flag_</b> - reprezentuje zmienną bool, która jest przypisywana do każdego zdarzenia czasowego. Za jej pomocą weryfikuje się czy zdarzenie ma być usunięte z listy zdarzeń
EndConsumption	Klasa reprezentująca zdarzenie czasowe: koniec konsumpcji klienta przy stoliku	<b>event_time_</b> - reprezentuje symulacyjny czas występowania zdarzenia <b>flag_</b> - reprezentuje zmienną bool, która jest przypisywana do każdego zdarzenia czasowego. Za jej pomocą weryfikuje się czy zdarzenie ma być usunięte z listy zdarzeń
EndDrinks	Klasa reprezentująca zdarzenie czasowe: koniec obsługi klientów przez kelnera	<b>event_time_</b> - reprezentuje symulacyjny czas występowania zdarzenia <b>flag_</b> - reprezentuje zmienną bool, która jest przypisywana do każdego zdarzenia czasowego. Za jej pomocą weryfikuje się czy zdarzenie ma być usunięte z listy zdarzeń
EndMeal	Klasa reprezentująca zdarzenie czasowe: koniec obsługi klientów przez kelnera	<b>event_time_</b> - reprezentuje symulacyjny czas występowania zdarzenia <b>flag_</b> - reprezentuje zmienną bool, która jest przypisywana do każdego zdarzenia czasowego. Za jej pomocą weryfikuje się czy zdarzenie ma być usunięte z listy zdarzeń
NewClient	Klasa reprezentująca zdarzenie czasowe: pojawienie się nowego klienta	<b>event_time_</b> - reprezentuje symulacyjny czas występowania zdarzenia <b>flag_</b> - reprezentuje zmienną bool, która jest przypisywana do każdego zdarzenia czasowego. Za jej pomocą weryfikuje się czy zdarzenie ma być usunięte z listy zdarzeń

TakeTableSit	Klasa reprezentująca zdarzenie czasowe: pojawienie się nowego klienta	<b>event_time_</b> - reprezentuje symulacyjny czas występowania zdarzenia <b>flag_</b> - reprezentuje zmienną bool, która jest przypisywana do każdego zdarzenia czasowego. Za jej pomocą weryfikuje się czy zdarzenie ma być usunięte z listy zdarzeń
--------------	---	---

### 3.Opis przydzielonej metody symulacyjnej

Schemat blokowy pętli głównej





## Lista zdarzeń czasowych i warunkowych

### Zdarzenia czasowe

Zdarzenie	Opis	Algorytm
Pojawienie się nowego klienta	Zdarzenie generuje nowego klienta i umieszcza ich w kolejce do bufetu lub w kolejce do kierownika.	<ol style="list-style-type: none"> <li>1. Wylosuj czas z generatora</li> <li>2. Utwórz nowego klienta ( restauracja lub bufet )</li> <li>3. Umieść klienta na koniec kolejki</li> <li>4. Zaplanuj przybycie nowego klienta</li> </ol>
Posadzenie klientów przy stoliku przez kierownika	Kierownik prowadząc klientów przez s czasu umieszcza ich w odpowiednim stoliku.	<ol style="list-style-type: none"> <li>1. Zwolnij kierownika</li> <li>2. Umieść klientów na końcu kolejki po napoje</li> </ol>
Koniec Konsumpcji w Bufecie	Zdarzenie wywołwane podczas zakończenia konsumpcji w bufecie przez określoną grupę klientów.	<ol style="list-style-type: none"> <li>1. Zwiększ liczbę dostępnych miejsc w bufecie o liczbę osób w grupie klientów</li> <li>2. Usuń klientów z bufetu</li> <li>3. Umieść klientów na końcu kolejki do kasjera</li> </ol>
Zakończenie obsługi przez kelnera ( napoje )	Kelner podał napoje klientom i zakończył obsługę	<ol style="list-style-type: none"> <li>1.Usuń klientów z grupy obsługiwanych przez kelnera</li> <li>2.Umieść klientów na koniec kolejki do obsługi przez kelnera ( jedzenie )</li> <li>3. Zwiększ liczbę dostępnych kelnerów o 1</li> </ol>
Zakończenie obsługi przez kelnera ( jedzenie )	Kelner podał jedzenie klientom i zakończył obsługę	<ol style="list-style-type: none"> <li>1.Usuń klientów z grupy obsługiwanych przez kelnera</li> <li>2.Umieść klientów do grupy konsumpcja</li> <li>3. Zwiększ liczbę dostępnych kelnerów o 1</li> </ol>
Zakończenie konsumpcji w stoliku	Klient zakończył konsumpcję w stoliku i udaje się do kolejki do kasy	<ol style="list-style-type: none"> <li>1.Usuń klientów z grupy konsumpcja</li> <li>2.Zwiększ liczbę dostępnych stolików</li> <li>3.Umieść klientów w kolejce do kasy</li> </ol>
Zakończenie obsługi przez kasjera	Kasjer zakończył obsługę klienta	<ol style="list-style-type: none"> <li>1.Zwiększ liczbę dostępnych kasjerów o 1</li> </ol>
Alarm	W restauracji zamontowano nieprawidłowo system przeciwpożarowy. Co jakiś czas, bez przyczyny, rozlega się dźwięk alarmu.	<ol style="list-style-type: none"> <li>1. Wylosuj czas z generatora</li> <li>2. Usuń z restauracji wszystkich nieświadomych klientów i wszystkie przypisane do nich zdarzenia czasowe</li> <li>3. Zaplanuj czas następnego alarmu</li> </ol>

## Zdarzenia warunkowe

Zdarzenie	Opis	Algorytm
Przydzielenie stolików ManagerWork	Kierownik zaczyna prowadzić klientów z kolejki do stolika	1.Usuń klientów z kolejki do stolików 2.Zmniejsz liczbę dostępnych stolików 3.Zaplanuj zdarzenie czasowe: Posadzenie klientów przy stoliku przez kierownika
Przydziel stanowisko w bufecie BuffetMembers	Umieszczenie klientów z kolejki w bufecie	1.Usuń klientów z kolejki do bufetu 2.Zmniejsz liczbę dostępnych miejsc w bufecie 3.Zaplanuj czas końca konsumpcji przez klientów
Zacznij obsługę klientów przez kelnera ( napoje ) WaiterDrink	Kelner rozpoczyna obsługę klientów	1.Usuń klienta z kolejki do kelnera 2.Zmniejsz liczbę dostępnych kelnerów o 1 3. Umieść klienta do grupy obsługiwanych przez kelnerów 4.Zaplanuj czas zakończenia obsługi
Zacznij obsługę klientów przez kelnera ( jedzenie ) WaiterFood	Kelner rozpoczyna obsługę klientów	1.Usuń klienta z kolejki do kelnera 2.Zmniejsz liczbę dostępnych kelnerów o 1 3. Umieść klienta do grupy obsługiwanych przez kelnerów 4.Zaplanuj czas zakończenia obsługi
Rozpocznij płatność u kasjera CashierWork		1.Usuń klientów z kolejki do kasjera 2. Zmniejsz liczbę dostępnych kasjerów 3.Zaplanuj czas zakończenia obsługi

## 4. Parametry wywołania programu

```
int Generators::avg_a_ = 420;  
int Generators::sigma_a_ = 50;  
  
int Generators::avg_b_ = 2900;  
int Generators::sigma_b_ = 80;  
  
int Generators::lambda_n_ = 830;  
  
int Generators::lambda_j_ = 2000;  
  
int Generators::lambda_f_ = 2020;  
  
int Generators::lambda_p_ = 2500;  
  
int Generators::avg_e_ = 4200;  
int Generators::sigma_e_ = 50;
```

## 5. Generatory

### Generowanie ziaren

Generowałem ziarna za pomocą funkcji:

$$h = x/q$$
$$x = a \cdot (x - q \cdot h) - r \cdot h$$

dla danych:

a=16807,  
q=127773,  
r=2836.

Po wygenerowaniu 10000 ziaren zapisałem do pliku seed.txt co 1000 ziarno.

Wygenerowane ziarna:

423669749	2025999803	169051120622	1164489620	370389800
2060109426	736449719	1623301720	125633416	105704401

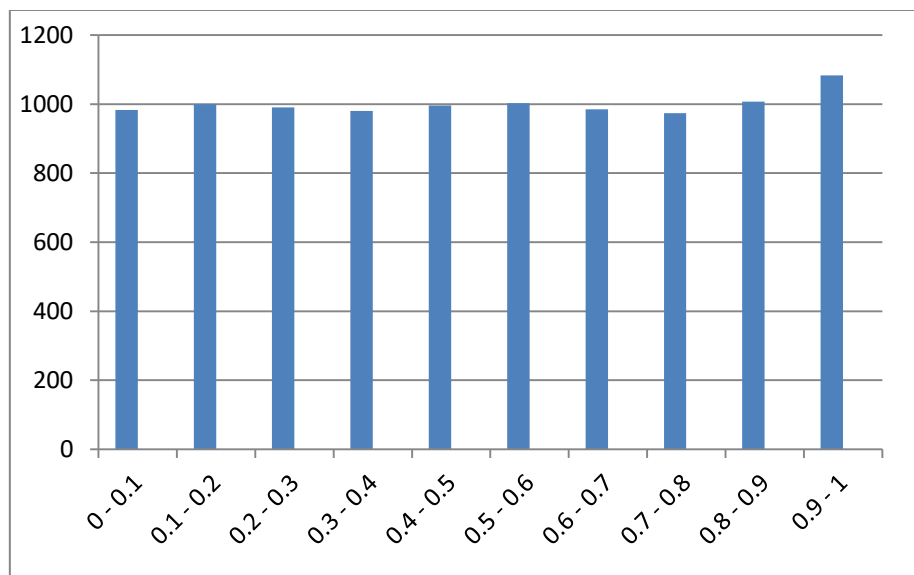
## Generator liczb pseudolosowych o rozkładzie równomiernym

Generator uzyskuje wartości w przedziale od 0 do 1

```
double Generators::Uniform(int & x)
{
    int a = 16807;
    int q = 127773;
    int r = 2836;
    double const range = 2147483647;    //2^31 -1

    int h = x / q;
    x = a * (x - q * h) - r * h;
    if (x < 0) x += range;
    double val = static_cast<double>(x) / range;
    return val;
}
```

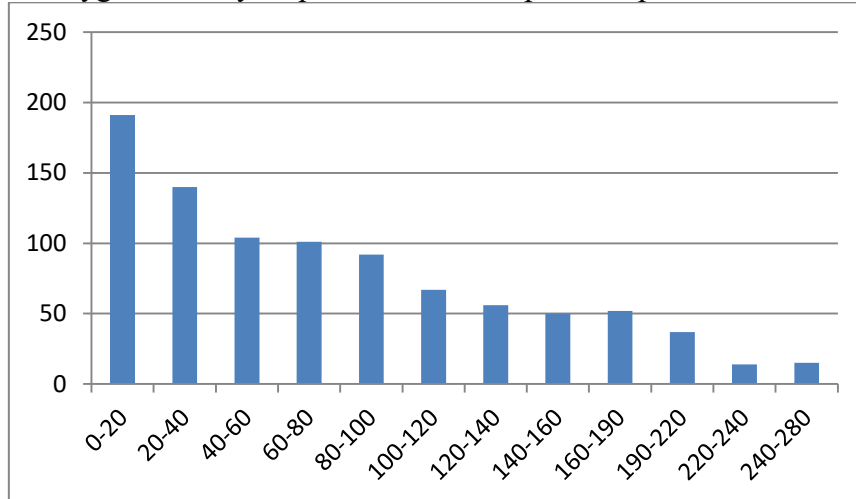
Histogram został wygenerowany na podstawie 10000 próbek



## Generator licz losowych o rozkładzie wykładniczym

```
double Generators::Exp(double avg, int & x)
{
    double k = log(Generators::Uniform((x)));
    double const val = ((-1)*(k))*avg;
    return val;
}
```

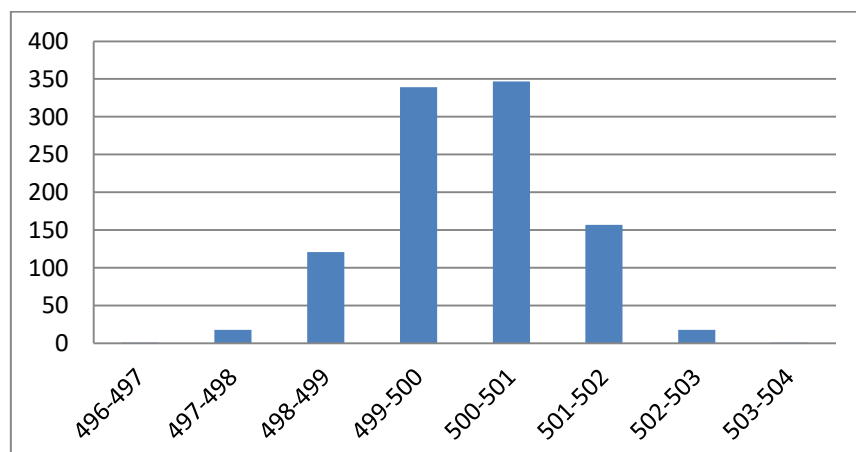
Histogram został wygenerowany na podstawie 10000 próbek i parametru  $\lambda = 200$



## Generator liczb losowych o rozkładzie normalnym

```
double Generators::Norm(const int average, const double variance, int & x)
{
    double X = 0;
    const int n = 12;
    for (auto i = 0; i < n; i++)
    {
        X += Uniform(x);
    }
    return int(((X - (double(n) / 2))*(variance * 12 / n)) + average);
}
```

Histogram został wygenerowany na podstawie 10000 próbek i parametrów:  $\mu = 500$  oraz  $\sigma = 1$



## Niezależność sekwencji losowych

Najistotniejszą metodą zapewnienia niezależności jest fakt, że każda z symulacji rozpoczyna się z innym ziarnem początkowym, którym jest czas w milisekundach.

Generator liniowy wykorzystuje wartości  $q = 127773$  i  $r = 2836$  co pozwala uzyskać  $2^{31} - 1$  niepowtarzalnych wartości

## 6.Opis zastosowanej metody testowania i weryfikacji poprawności działania programu.

Weryfikacja poprawności działania programu polegała m.in. na krokowym przechodzeniu przez pętle symulacyjną,

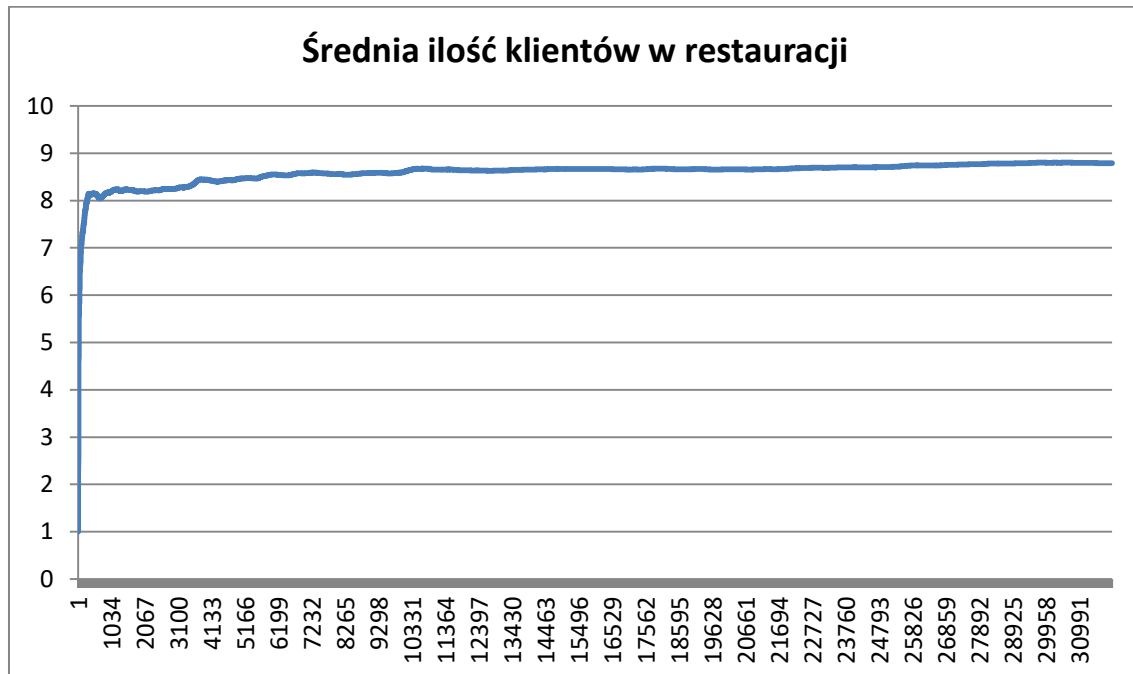
W ten sposób sprawdzałem czy zdarzeniom czasowym przypisywane są odpowiednie czasy symulacyjne, czy generatory zwracają odpowiednie wartości do ich parametrów oraz czy sortowanie listy zdarzeń przebiega prawidłowo.

Pomocnym sposobem weryfikacji było także debugowanie metod zastosowanych w symulacji.

Pozwoliło to sprawdzić i udowodnić, że zastosowane algorytmy, np. wyboru z kolejki pierwszej grupy mieszczącej się w wolnych stolikach, działają prawidłowo.

## 7. Wyniki symulacji

Wyznaczenie długości fazy początkowej



Wykres powstał na podstawie 10 symulacji z ziarnami z pliku seed.txt

Wykres stabilizuje się przy wartości około 5000 klientów i taką wartość przyjąłem jako koniec fazy początkowej.

## Wyznaczenie parametrów symulacji

Podczas testowania symulacji konieczna była zmiana kilku parametrów.

Parametr	Wartość początkowa	Wartość końcowa
k	13	5
$n_2, n_3, n_4$	<b>8,14,4</b>	<b>8,14,4</b>
$p_1, p_2, p_3, p_4$	0.11,0.33,0.33,0.23	0.19,0.35,0.30,0.16
$\mu_a (\sigma_a^2)$	1500(100 <sup>2</sup> )	420(50 <sup>2</sup> )
s	40	240
b	<b>20</b>	<b>20</b>
$\mu_b (\sigma_b^2)$	<b>2900(80<sup>2</sup>)</b>	<b>2900(80<sup>2</sup>)</b>
$\lambda_n$	370	830
$\lambda_j$	2000	2000
$\lambda_f$	<b>2020</b>	<b>2020</b>
c	<b>6</b>	<b>6</b>
$\lambda_p$	220	2500
$\mu_e (\sigma_e^2)$	<b>4200(50<sup>2</sup>)</b>	<b>4200(50<sup>2</sup>)</b>

Największym zmianom uległy parametry:

- Ilość kelnerów w symulacji
- Średnia i wariancja pojawiania się nowych klientów
- Średni czas obsługi przez kasjera

Tabela z wynikami symulacji dla każdego przebiegu symulacyjnego

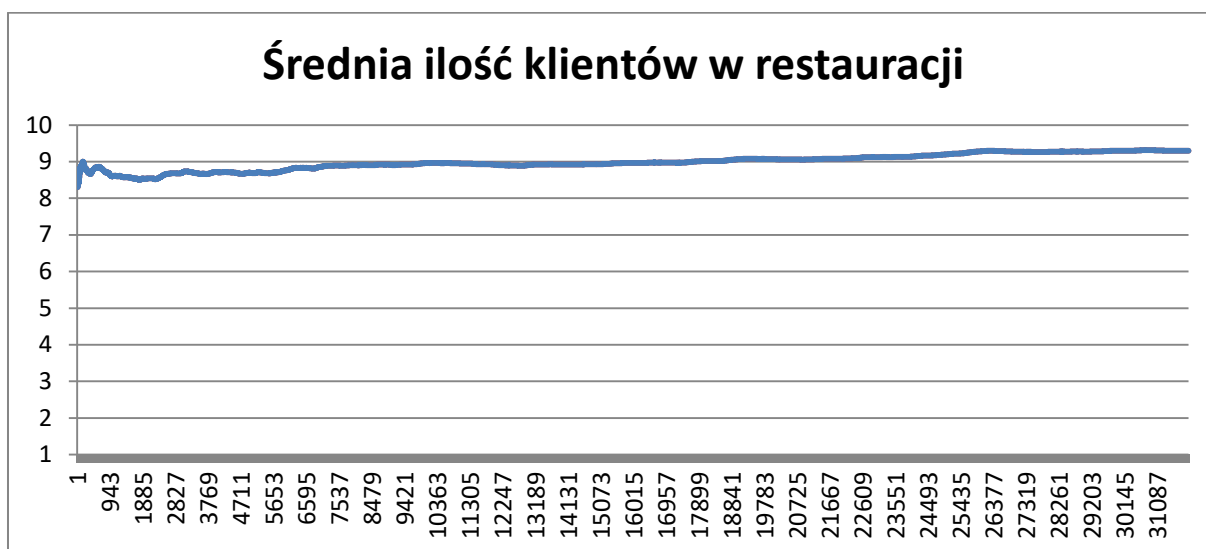
nr symulacji	średni czas oczekiwania na stolik	średnia długość kolejki oczekujących na stolik	średni czas oczekiwania na kelnera	średnia długość kolejki przy kasach
1	571,2333	0,996202	8,253636	0,425164
2	1512,058	2,090308	9,058175	0,568700
3	1094,51	1,609844	14,91208	0,440516
4	131,8122	0,491381	23,15778	0,523892
5	1726,533	2,336981	5,461877	0,422959
6	94,69939	0,480836	29,50044	0,561609
7	132,9122	0,534596	20,71463	0,469176
8	203,8316	1,437249	20,4848	0,398151
9	214,9111	0,593854	23,58792	0,444952
10	131,7714	0,491381	14,91208	0,561609



Wyniki końcowe w postaci uśrednionych wyników po wszystkich przebiegach

średni czas oczekiwania na stolik	średnia długość kolejki oczekujących na stolik	średni czas oczekiwania na kelnera	średnia długość kolejki przy kasach
581,427223	1,106263	17,004312	0,481673

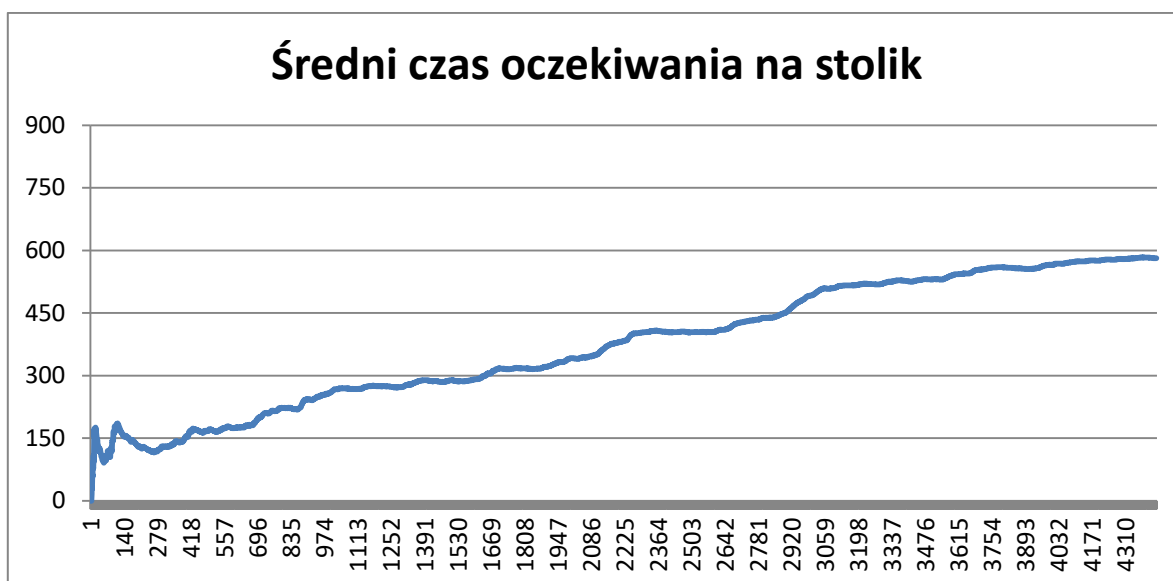
Wykres średniej ilości klientów w restauracji po odcięciu fazy początkowej



Wykres powstał na podstawie uśrednienia danych z 10 symulacji z ziarnami z pliku seed.txt

Po odcięciu fazy początkowej na wykresie od początku stabilizuje się na poziomie 9 klientów.

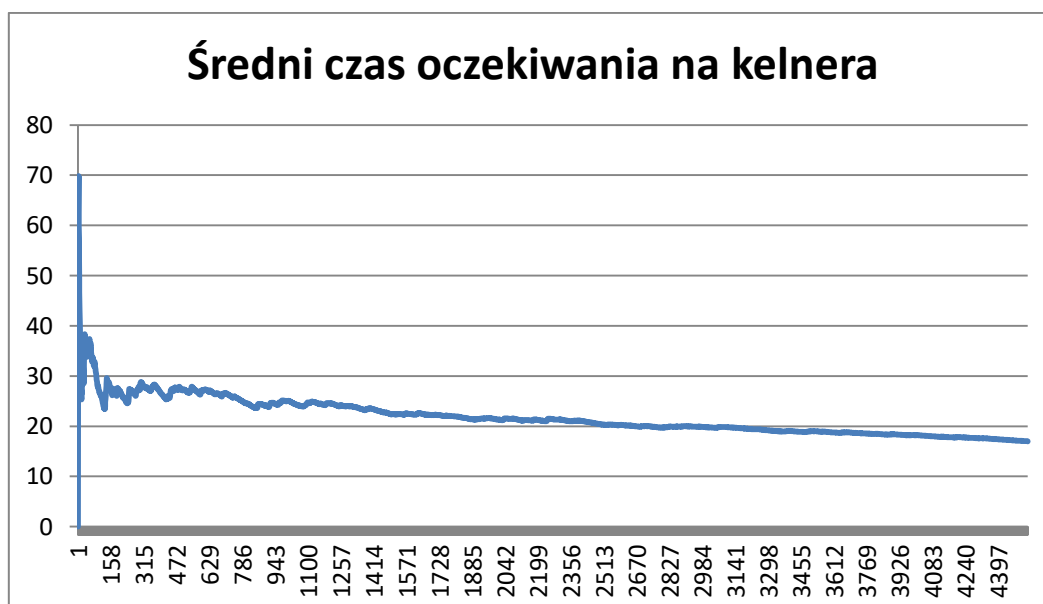
Wykres średniego czasu oczekiwania na stolik po odcięciu fazy początkowej



Wykres powstał na podstawie uśrednienia danych z 10 symulacji z ziarnami z pliku seed.txt

Po analizie symulacji stwierdziłem, że wartość średniego czasu oczekiwania na stolik jest mocno skorelowana z ilością grup 4 osobowych klientów w kolejce, ponieważ to ta grupa ma najmniejsze szanse na wejście do stolika.

Wykres średniego czasu oczekiwania na kelnera po odcięciu fazy początkowej



Wykres powstał na podstawie uśrednienia danych z 10 symulacji z ziarnami z pliku seed.txt

Jak można zauważyć na wykresie średni czas oczekiwania na kelnera stabilizuje się na poziomie 17 czasu symulacyjnego.

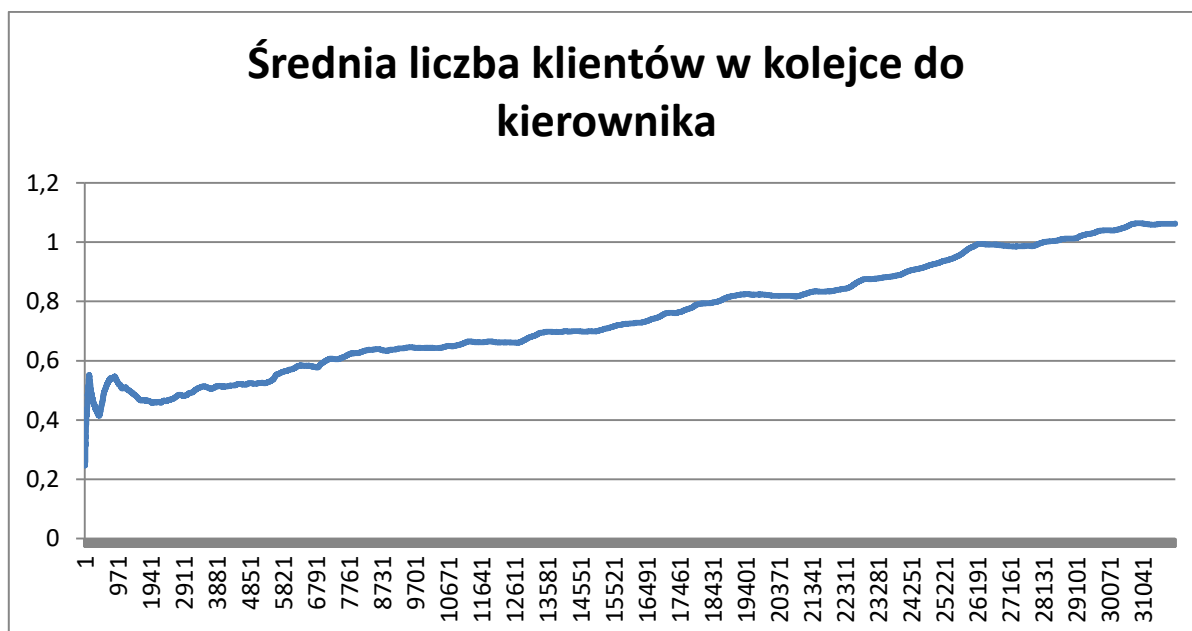
Wykres średniej kolejki do kasjerów po odcięciu fazy początkowej



Wykres powstał na podstawie uśrednienia danych z 10 symulacji z ziarnami z pliku seed.txt

Po zwiększeniu czasu obsługi przez kasjera można zauważyć stabilizację ilości osób w kolejce.

Wykres średniej kolejki do kierownika po odcięciu fazy początkowej



Wykres powstał na podstawie uśrednienia danych z 10 symulacji z ziarnami z pliku seed.txt

Po odpowiednim doborze parametrów pojawiania się nowego klienta i parametru s wykres stabilizuje się na poziomie 1 klienta w kolejce.

Przedziały ufności dla parametrów z prawdopodobieństwem 95%

Parametr	min	średnia	max
<b>średni czas oczekiwania na stolik</b>	191,6168	581,4272236	971,2376
<b>średnia długość kolejki oczekujących na stolik</b>	0,663402	1,106263	1,549124
<b>Średni czas oczekiwania na kelnera</b>	12,18152	17,00434112	21,82717
<b>średnia długość kolejki przy kasach</b>	0,440899	0,481673	0,522447

## 8.Wnioski

Otrzymane wyniki symulacyjne są zadowalające. Poprawnie wyznaczyłem parametr końca fazy początkowej co widać na wykresie średniej liczby osób w systemie po jej odcięciu. Po wielu próbach zmian parametrów symulacji jedynym parametrem, którego nie udało mi się ustabilizować jest średni czas oczekiwania na stół. Wynika to z faktu dłuższego, w stosunku do mniej licznych grup, oczekiwania na stół przez 4 osobowe grupy klientów, którzy mogą usiąść jedynie w stole 4 osobowym.

Model symulacyjny może zostać wykorzystany w rzeczywistej restauracji w celu optymalizacji jej parametrów np. ilości stołów.